myClicker

Identification:

**Project: myClicker**

**Project Manager:** Rachael Batchelder (rbatchel@princeton.edu)

**The A-Team:**

Margaret Tam (mtam@princeton.edu)

Max Rubin (merubin@princeton.edu)

Timothy Trieu (ttrieu@princeton.edu)

Yingxue Li (yingxuel@princeton.edu)

Overview:

myClicker is a web-based question-and-response application for class use, which was inspired by the already existing iClicker. The iClicker is a device that allows students to select buttons A, B, C, D, or E in response to a multiple choice question proposed by the professor., which can then be vi ewed through the use of specific software.  Our application seeks to expand and improve upon this functionality by by providing a website that allows for free-form responses by the students. Additionally, questions can be submitted by the students to the professor, and both questions and responses can be submitted anonymously. This application will hopefully encourage greater participation from students who may be shy about asking or answering questions in class. At the same time, it will prove to be a useful tool for professors who can better gauge the comprehension of their students by reviewing whatever questions they may have, as well as by asking more complex questions than achievable with the iClicker system.

The central function of the myClicker is to enable the submission of questions and responses as described above. In order to support this function, the application will also have features like the option of anonymity and processing of simple responses to generate a suitable, easily digestible display for the professor. In order to make reviewing students’ questions easier for the professor, the application will also allow students to “support” questions asked by other students so that more popular questions can swim to the top of the pool. Student questions and responses will also be “tagged” by a student id, and can therefore be sorted and tracked by the professor.  
  
Since the available features and information differs greatly depending on the type of user (professor or student), there will be separate interfaces for each one as designated by their role in each class. Since the submission and organization of student questions resembles that of Piazza, the format and structure of the student side will be fairly similar, with a taskbar to the left filled with the submitted questions, and an additional “thumbs-up” icon beside each one. The professor’s interface will be focused more on opening and closing questions with a tab for opening/drafting questions, a taskbar on the left for viewing past questions, a button for each open question to close it, and a tab for viewing student/question statistics.

Functionality:

*Front Page*

The front page of the myClicker website  will be dominated by a sign-in panel, with an option to sign up as a new user (using netID through the Central Authentication Service). There will be miscellaneous informative blurbs about how myClicker works.

*Personal Page*

The left side of a user’s personal page offers registration options to open or enlist in classes. Professors can open a new “slot” for their class, which updates the database. They can then populate the class description with relevant information such as location, times, and contact information. Above the “Open a New Class” panel will be a “Find a Class” search bar, where students can enter a course name, professor, or course abbreviation to search for open classes. Students can then select the class and sign up using net ID. Designation as professor or student will be assigned during registration of a new class, according to whether the user opened the class or signed up for it.

*Student Signed In*

Users that are student members of a class can go to the class session webpage through an option on their personal page.  This webpage will have a question taskbar to the left, and central area dominated by a single “responses” section, which will contain some of the recent activity of the class, including posted questions and answers.   At any point during the lecture, students can type and post responses (anonymously or by id) to questions the professor makes in class.  The user interface will automatically reload a section that displays recent responses when new ones are submitted, so the professor can see student responses in real time.   These question sets will be logged in the website’s database, allowing users to view previous responses.  Likewise, students can also post questions to the question taskbar, which will also be visible to all students.  Students that agree with a question can designate that they “support” it, reducing the need for redundant questions. There will also be a confusometer for students, so if students do not understand the current material, they can increase their meter and if there are sufficient confused students, the professor will receive an alert in real time.

*Professor Signed In*

Professors will be able to open and close questions either during lecture or use the draft question feature to write the question/possible answers before lecture. After a question is opened, or released, students will be able to submit their individual answers, which will be accessible by the professor. The professor also has the option to view the responses for each question or statistics for each student/question (ie. how many questions has a student answered, time and date of each response) as well as the ability to search open ended responses by keywords or sort them by their similarities. After closing a question, students are unable to submit answers.

Design:

The user interface will be written in Ajax.  Due to the frequent comments from students, it can be expected that several fields for the interface will change in a matter of seconds, so rapid updates to pages will be required for this tool to be effective.  
  
Most of the website’s processing will occur on the server side.  This will allow the server to update all connected users simultaneously.  The server code will be written in Django, and will handle tasks such as sending updates to the connected users, parsing and processing student questions and responses, and creating and maintaining class slots.  
  
The database of the website will be written in MySQL.  It will contain data pertaining to the class slots that have been created, the registered users (along with their activity and status), and the questions and responses submitted by the students.  The website itself will be hosted on Heroku.

Risks and Open Issues:

* No team members have experience with web development or server hosting.
* Need to handle users separately depending on their status as student or professor (it may be possible for a user to be both)
* Parsing responses and displaying them in organized format.
* Make a simple user interface so that professors will be likely to use it.

Projected Timeline/Milestones:

March 23, 2013  
    Familiarize with appropriate languages over Spring Break.  Contact professors to arrange meetings to discuss their interest in design features and field testing the website.  
  
March 31, 2013  
    Establish basic website server (not necessarily functional, also GitHub)  
  
April 12, 2013

Project prototype.  Demonstration of submission of questions to the website and displaying to multiple users.  
  
April 26, 2013  
    Alpha test.  Single class session with students and professor participation.  
  
May 3, 2013  
    Beta test.  Transition of a user between multiple sessions and use of the student question section.  
  
May 8-10, 2013  
    Demo for the class  
  
May 11, 2013  
    Project Demo  
  
May 14, 2013  
    Submit final project  
  
**Group timeline:**  
  
[http://www.princeton.edu/~rbatchel/timeline.html](http://www.princeton.edu/~rbatchel/timeline.html%20)